

US-PAT-NO: 5557659

DOCUMENT-IDENTIFIER: US 5557659 A

**\*\*See image for Certificate of Correction\*\***

TITLE: Electronic mail system having integrated voice messages

----- KWIC -----

US Patent No. - PN (1):

5557659

Detailed Description Text - DETX (17):

FIG. 5 is a flowchart describing the steps for message playback of new messages of a voicemail system integrated with MAPI E-mail systems. In 501, the user name and password is provided to the MAPILogon subroutine. The MAPIFindNext subroutine is used to get the next mail message, step 502. Next, the MAPIReadMail subroutine is used to get the information about that message, step 503. A determination is made as to whether this is a new message, step 504. If so, a determination is then made as to whether there are any voicemail attachments, step 505. If it is determined that there are voicemail attachments, the .vox files are extracted from the message, step 506.

Detailed Description Text - DETX (18):

The extracted .vox file is played by using the message playback options, step 507. If it is determined in steps 504 and 505 that either this is not a new message or that there are no voicemail attachments, step 508 is performed immediately. If it is determined in step 508 that there are no more messages, the MAPILogoff subroutine is used to end the session, step 509. Otherwise, step 502 is repeated to get the next mail message. Note that this flowchart shows message playback for new messages.

Detailed Description Text - DETX (19):

FIG. 6 is a flowchart showing the steps for playback of saved or old messages. In step 601, the user name and password is provided to the MAPILogon subroutine. The MAPIFindNext subroutine is used to get the next mail message, step 602. Next, the MAPIReadMail subroutine is used to get the information about that message, step 603. A determination is made as to whether this is a saved or old message, step 604. If so, a determination is then made as to whether there are any voicemail attachments, step 605. If it is determined that there are voicemail attachments, the .vox files are extracted from the message, step 606.

Detailed Description Text - DETX (20):

The extracted .vox file is played by using the message playback options, step 607. If it is determined in steps 604 and 605 that either this is not a saved message or that there are no voicemail attachments, step 608 is performed immediately. If it is determined in step 608 that there are no more messages, the MAPILogoff subroutine is used to end the session, step 609. Otherwise, step 512 is repeated to get the next mail message.

Detailed Description Text - DETX (29):

A determination is made as to whether the attachment is a .vox file, step 1206. If so, the VIMGetMessageItem subroutine is used to extract the .vox file from the message, step 1207. The extracted .vox file is played back by using the message playback options, step 1208. If it is determined in step 1206 that the attachment is not a .vox file, then steps 1207 and 1208 are skipped and the determination of step 1209 is performed. If there are more attachments in the message, step 1205 is repeated. Otherwise, the VIMCloseMessage subroutine is used to close this mail message, step 1210. In step 1211, a determination is made as to whether there are any more messages. If so, step 1202 is repeated. Otherwise, the VIMCloseMessageContainer subroutine is used to close the Inbox of the caller, step 1212. Not shown for VIM is the play back of saved messages. As for MAPI shown in FIG. 5b, it simply changes step 1204 in FIG. 12 to look for "old/saved" messages instead of new.

Detailed Description Paragraph Table - DETL (1):

TABLE 1

---

What's happening MAPI call VIM API call

---

PC/Workstation Software Operations Record a message and MAPISendMessage then press the mail button to Listen to voice No special API messages found in function required - function required - the user's mail autoloads special autoloads special folder/In Tray Inbox application on the application on the K based on K based on association to .VOX association to .VOX extension extension General Housekeeping Administration Assign voicemail A template that Use the cc:Mail Admin box numbers has two additional (DOS) program to create and passwords; fields aliases for existing also default 'Voice' mailbox users. These aliases telephone Number' and contain the mailbox extensions 'Default Telephone number as the name, required for Extension' is the telephone extension 'Connect to defined. In the comments field Sender'. The standard Mail and user name in the address book form address field. has two 'Phone Number' fields. The 'Phone number #2 is used as the Voice mailbox number, and Phone number #1 is the extension number. Voice Gateway A text file VIMGetDefaultSession! User Table containing the nfo Creation. The MAPI address book VIMOpenSession User Table is created using the VIMOpenAddressBook created MS-Mail VIMEnumerateAddress periodically and template.exe BookEntries is updated every utility, which is VIMCloseAddressBook time the Voice read into memory. VIMCloseSession

Gateway is This contains the restarted. required information about the  
 users. General Session InitMAPI VIMInitialize Management MAPILogon  
 VIMOpenSession MAPILogoff VIMCloseSession DelnitMAPI VIMTerminate PBX  
 Integrated Call Answering Call forwards to Look up the Look up the extension  
Voice Gateway extension number in the user table with called number in the  
 created at start-up to get extension user table E-mail Mailbox number. number  
 created at start- information up to get E-mail Mailbox number. External  
 caller MAPILogon VIMGetDefaultSessionInfo records message MAPISendMail  
 VIMOpenSession for the called MAPILogoff VIMCreateMessage extension user.  
 VIMSetMessageHeader Need to send VIMSetMessageRecipient voice file to the  
 VIMSendMessage users E-mail VIMCloseSession address. User Calls In to Listen  
 to Messages User calls Voice Check name Check name from user Gateway to from  
 user table. table. Check password check voice Check password using  
 VIMOpenSession messages. User using enters Voice MAPILogon mailbox number  
 and password. Count and MAPIFindNext VIMOpenMessageContainer access voice  
 MAPIReadMail VIMEnumerateMessages files. (look for vox VIMOpenMessage files)  
 VIMEnumerateMessageItem MAPIFreeBuffer ms (look for vox files) MAPILogoff  
 VIMGetMessageItem VIMCloseMessage VIMCloseMessageContainer VIMCloseSession  
 Tell User how All.vox files All.vox files found above many voice found above  
 have been extracted from messages. have been their messages and their  
extracted from filenames are sent to the their messages Voice gateway PC for  
 and their playing back. filenames are sent to the Voice gateway PC for  
 playing back Listen to MAPIReadMail VIMOpenMessageContainer messages (without  
 the VIMMarkMessageAsRead MAPI.sub.- PEEK VIMCloseMessageContainer flag)  
 MAPIFreeBuffer Message Playback Options Options during Message Playback  
 Connect to message The Voice Server The Voice Server has sender has received  
 the received the telephone telephone extension extension of the of the  
 sender with sender with the the other information other information of the  
 vox file of the .vox file Reply to sender Combinations of Combinations of  
 above above Forward to another Combinations of Combinations of user above  
 above Save MAPI SaveMail is VIMMarkMessageAs used to Save a mail Read  
 message Record a new No special API No special API prompt functions required  
 functions required Delete MAPIReadMail VIMOpenMessageCon MAPIDeleteMail  
 tainer MAPIFreeBuffer VIMRemoveMessage VIMCloseMessageCon tainer Cycle  
 forwards and No Special No special functions backwards through functions  
 required required messages

---

US-PAT-NO: 5771355

DOCUMENT-IDENTIFIER: US 5771355 A

TITLE: Transmitting electronic mail by either reference or  
value at file-replication points to minimize costs

----- KWIC -----

US Patent No. - PN (1):  
5771355

Detailed Description Text - DETX (53):

As will be understood, the URL 627 which is sent as the message pointer may be either simple or very complex. For example, it may be as simple as a particular attachment file which may be retrieved, such as a text file or spreadsheet. Alternatively, the URL may instead be a pointer to an HTML document itself at another location accessible by or visible to users of the WWW. As will be appreciated, an advantage of using such complex URLs is that an "attachment" may be contextually included in the body of an e-mail message, rather than merely attached as an available file for viewing with less context. For instance, the subject or longer e-mail primary message may be left, having a hypertext link based on the URL to a spreadsheet. The words "our financial analysis" could be incorporated as a hypertext link into the subject line or into a longer text e-mail message left on the recipient's mail page. Then, instead of reading the text and then opening the attached spreadsheet (with some concomitant loss of context), as in current e-mail systems, the user could click on the reference itself while reading the message, thus improving the textual reference.

US-PAT-NO: 5818447

DOCUMENT-IDENTIFIER: US 5818447 A

TITLE: System and method for in-place editing of an electronic  
mail message using a separate program

----- KWIC -----

US Patent No. - PN (1):  
5818447

Brief Summary Text - BSTX (16):

More particularly described, the message includes an attachment list. The container object displays the attachment data in the body region, with the attachment data being rendered by the program module in association with the body data. The attachment data is extracted from the program module and stored in the attachment list portion of the message item. In some cases, formatting codes included in the body data are removed prior to storing the body data in the message item. Similarly, the program module may be provided with template identification data corresponding to formatting to be applied to the body data.

Detailed Description Text - DETX (2):

The present invention is directed to an improved system and method for editing and viewing an email message. Generally described, an embodiment of the present invention employs a Document Object-enabled mail note or form to provide a Document Object container with an email-related user interface (UI) and a view port or frame for viewing and editing the contents of an email message. Instead of implementing text editing or word processing functionality in the mail note itself, the mail note allows a separate, full-featured word processing program to display and edit the email message in the view port provided by the mail note. The mail note then extracts the message data from the word processor and reformats the data to comply with the format required by the email client program. This allows the user to edit and view email messages using the editing environment and formatting capabilities of the full-featured word processing program, while also providing messages that are compatible with various types of email clients.

Detailed Description Text - DETX (120):

At step 1355 the container mail note extracts the message data from the word processor document object. This is accomplished using the IMsoMailSite, IMsoFormSite, and IMsoMailEditor interfaces described above. As the message data is extracted, it is provided in both RTF and plain text data streams. In

the plain text data stream, some special characters and formatting is reformatted to ensure that it may be viewed on downlevel mail clients. Attachments are provided separately to the container mail note. In addition, attachments are removed from the body of the message and stored in the attachments portion of the MAPI message store.

Detailed Description Text - DETX (133):

Furthermore, the messages created in this manner can be opened and read by any MAPI compatible email client. This is accomplished by ensuring that the container mail note converts the word processor's formatting conventions into a form that is consistent with MAPI. In addition to the handling of RTF text, this requires that embedded attachments are removed from the text of the message and separately stored in the message store.

US-PAT-NO: **5754306**

DOCUMENT-IDENTIFIER: US 5754306 A

TITLE: System and method for a communication system

----- KWIC -----

US Patent No. - PN (1):

**5754306**

Brief Summary Text - BSTX (20):

In that embodiment, several methods are provided for receiving a package from the electronic mail provider. One method extracts the message from the package, another method extracts the distribution list, and a third method extracts any file attachments.

Detailed Description Text - DETX (132):

When a package is received from a communication service, a method DestroyData strips from a package received Service Specific data. A method "GetNote" retrieves from a package an included message. A method "GetDistList" retrieves from the package a list of recipients and casts the retrieved list of recipients into a Group type address card. Attachment files are retrieved from a package using the method "GetAttachments", which returns a list of strings containing the file names of the attachments.

US-PAT-NO: 5717742

DOCUMENT-IDENTIFIER: US 5717742 A

**\*\*See image for Certificate of Correction\*\***

TITLE: Electronic mail system having integrated voice messages

----- KWIC -----

US Patent No. - PN (1):

5717742

Detailed Description Text - DETX (16):

FIG. 5 is a flowchart describing the steps for message playback of new messages of a voicemail system integrated with MAPI E-mail systems. In 501, the user name and password is provided to the MAPILogon subroutine. The MAPIFindNext subroutine is used to get the next mail message, step 502. Next, the MAPIReadMail subroutine is used to get the information about that message, step 503. A determination is made as to whether this is a new message, step 504. If so, a determination is then made as to whether there are any voicemail attachments, step 505. If it is determined that there are voicemail attachments, the .vox files are extracted from the message, step 506.

Detailed Description Text - DETX (17):

The extracted .vox file is played by using the message playback options, step 507. If it is determined in steps 504 and 505 that either this is not a new message or that there are no voice mail attachments, step 508 is performed immediately. If it is determined in step 508 that there are no more messages, the MAPILogoff subroutine is used to end the session, step 509. Otherwise, step 502 is repeated to get the next mail message. Note that this flowchart shows message playback for new messages.

Detailed Description Text - DETX (18):

FIG. 6 is a flowchart showing the steps for playback of saved or old messages. In step 601, the user name and password is provided to the MAPILogon subroutine. The MAPIFindNext subroutine is used to get the next mail message, step 602. Next, the MAPIReadMail subroutine is used to get the information about that message, step 603. A determination is made as to whether this is a saved or old message, step 604. If so, a determination is then made as to whether there are any voicemail attachments, step 605. If it is determined that there are voicemail attachments, the .vox files are extracted from the message, step 606.



Detailed Description Text - DETX (19):

The extracted .vox file is played by using the message playback options, step 607. If it is determined in steps 604 and 605 that either this is not a saved message or that there are no voicemail attachments, step 608 is performed immediately. If it is determined in step 608 that there are no more messages, the MAPILogoff subroutine is used to end the session, step 609. Otherwise, step 512 is repeated to get the next mail message.

Detailed Description Text - DETX (28):

A determination is made as to whether the attachment is a .vox file, step 1206. If so, the VIMGetMessageItem subroutine is used to extract the .vox file from the message, step 1207. The extracted .vox file is played back by using the message playback options, step 1208. If it is determined in step 1206 that the attachment is not a .vox file, then steps 1207 and 1208 are skipped and the determination of step 1209 is performed. If there are more attachments in the message, step 1205 is repeated. Otherwise, the VIMCloseMessage subroutine is used to close this mail message, step 1210. In step 1211, a determination is made as to whether there are any more messages. If so, step 1202 is repeated. Otherwise, the VIMCloseMessageContainer subroutine is used to close the Inbox of the caller, step 1212. Not shown for VIM is the play back of saved messages. As for MAPI shown in FIG. 5b, it simply changes step 1204 in FIG. 12 to look for "old/saved" messages instead of new.

Detailed Description Paragraph Table - DETL (1):

TABLE 1	What's happening
VIM API call	MAPI call
Operations	Record a <u>message</u> MAPISendDocuments SMISendMail and then press
the <u>mail</u> button	Listen to <u>voice</u> No special API No special API <u>messages</u>
found	function required - function required - in the user's <u>mail</u> autoloads
special	autoloads special folder/In Tray application on the application on
the Inbox	PC based on PC based on association to .VOX association to .VOX
extension	extension General Housekeeping Administration Assign voicemail A
template that	Use the cc: <u>Mail</u> Admin box numbers has two additional (DOS)
program to create	and passwords; fields aliases for existing also default
<u>'Voice</u>	mailbox users. These aliases telephone Number' and contain the mailbox
extensions	'Default Telephone number as the name, required for Extension'
is the telephone	extension 'Connect to defined. in the comments field
Sender'.	The standard <u>Mail</u> and user name in the address book form address
field.	has two 'Phone Number' fields. The 'Phone number #2 is used as the
<u>Voice</u>	mailbox number, and Phone number #1 is the extension number. <u>Voice</u>
Gateway	A <u>text</u> file VIMGetDefaultSessionInfo User Table containing the
VIMOpenSession	Creation. The MAPI address book VIMOpenAddressBook User
Table is	is created using the VIMEnumerateAddress created MS- <u>Mail</u>
BookEntries	periodically and template.exe VIMCloseAddressBook is updated
every utility,	which is VIMCloseSession time the <u>Voice</u> read into memory.
Gateway is	This contains the restarted. required information about the
users.	General Session InitMAPI VIMInitialize Management MAPILogon

VIMOpenSession MAPILogoff VIMCloseSession DelnitMAPI VIMTerminate PBX  
Integrated Call Answering Call forwards to Look up the Look up the extension  
Voice Gateway extension number in the user table with called number in the  
created at start-up to get extension user table E-mail Mailbox number. number  
created at start- information up to get E-mail Mailbox number. External  
caller MAPILogon VIMGetDefaultSessionInfo records message MAPISendMail  
VIMOpenSession for the called MAPILogoff VIMCreateMessage extension user.  
VIMSetMessageHeader Need to send VIMSetMessageRecipient voice file to the  
VIMSendMessage users E-mail VIMCloseSession address. User Calls In to Listen  
to Messages User calls Voice Check name Check name from user Gateway to  
from user table. table. Check password check voice Check password using  
VIMOpenSession messages. User using enters Voice MAPILogon mailbox number  
and password. Count and MAPIFindNext VIMOpenMessageContainer access voice  
MAPIReadMail VIMEnumerateMessages files. (look for .vox VIMOpenMessage  
files) VIMEnumerateMessageItems MAPIFreeBuffer (look for .vox files)  
MAPILogoff VIMGetMessageItem VIMCloseMessage VIMCloseMessageContainer  
VIMCloseSession Tell User how All .vox files All .vox files found above  
many voice found above have been extracted from messages. have been their  
messages and their extracted from filenames are sent to the their messages  
Voice gateway PC for and their playing back. filenames are sent to the Voice  
gateway PC for playing back Listen to MAPIReadMail VIMOpenMessageContainer  
messages (without the VIMMarkMessageAsRead MAPI.sub.-- PEEK  
VIMCloseMessageContainer flag) MAPIFreeBuffer Message Playback Options  
Options during Message Playback Connect to The Voice Server The Voice  
Server has message sender has received the received the telephone telephone  
extension extension of the of the sender with sender with the the other  
information other information of the .vox file of the .vox file Reply to  
sender Combinations of Combinations of above above Forward to Combinations  
of Combinations of another user above above Save MAPI SaveMail is  
VIMMarkMessageAs used to Save a mail Read message Record a new No special  
API No special API prompt functions required functions required Delete  
MAPIReadMail VIMOpenMessageContainer MAPIDeleteMail VIMRemoveMessage  
MAPIFreeBuffer VIMCloseMessageContainer Cycle forwards No Special No special  
functions and backwards functions required required through messages

---